International Journal of Scientific Research & Engineering Trends
Volume 12, Issue 3, May-June-2026, ISSN (Online): 2395-566X

MedLens: An AI-Powered Radiology Report Simplification System for Improved Patient Accessibility

B. M. Promod Kumar[1], Bhavana N. S.[2], C. Chinmayi[2], Deepthi C. Shekar[2], Deenadayal B. K.[2]

[1]Assistant Professor, Department of Computer Science & Engineering, P.E.S. College of Engineering, Mandya – 571401, Karnataka, India
[2]Student, Department of Computer Science & Engineering, P.E.S. College of Engineering, Mandya – 571401, Karnataka, India

Abstract- Radiology reports generated from imaging modalities such as X-rays, Computed Tomography (CT), Magnetic Resonance Imaging (MRI), and ultrasound scans are critical clinical documents. However, these reports are authored in complex medical terminology intended for radiologists and specialist physicians, rendering them largely inaccessible to patients and non-medical users. This communication gap results in confusion, anxiety, and increased dependency on healthcare professionals for basic explanations. This paper presents MedLens, an AI- powered radiology report simplification system that bridges this gap by leveraging Natural Language Processing (NLP) and Retrieval-Augmented Generation (RAG). The system extracts text from uploaded PDF reports using PyMuPDF, processes clinical content using Google Gemini AI models, and generates accurate, context-aware patient-friendly summaries. It further classifies the urgency of findings into levels (Low, Moderate, High, Critical), and integrates multilingual translation, text-to-speech functionality, and an AI-powered contextual chatbot. The platform is deployed using FastAPI on the backend and React.js with Tailwind CSS on the frontend. Experimental results demonstrate that MedLens successfully simplifies complex medical terminology, detects critical conditions, provides multilingual support, and enables interactive report-based queries, thereby empowering patients with better health awareness and facilitating informed discussions with healthcare providers.

Keywords—Radiology report simplification, Natural Language Processing, Retrieval-Augmented Generation, Gemini AI, Patient accessibility, Urgency detection, Multilingual translation, Text-to-speech, Healthcare chatbot.

I. INTRODUCTION

The rapid digitisation of healthcare systems has enabled patients to access their diagnostic reports directly through electronic health portals and telemedicine platforms. Despite this progress, the gap between medical knowledge and patient comprehension remains wide. Radiology reports produced from CT scans, MRI scans, X-rays, and ultrasound imaging contain dense clinical terminology that is practically unintelligible to most patients [2].

The consequences include misinterpretation of findings, unnecessary anxiety, and delayed treatment decisions. Artificial Intelligence (AI), particularly NLP and large language models (LLMs), has demonstrated significant promise in transforming unstructured medical text into structured, patient-friendly content [3, 4]. Systems leveraging these technologies can extract key findings, classify severity, and generate accessible explanations without compromising clinical accuracy.

MedLens is proposed as a comprehensive AI-powered medical report assistant designed to address these challenges. The system accepts radiology and clinical PDF reports as input, extracts textual content using PyMuPDF-based OCR, and processes the extracted data through a multi-stage AI pipeline producing: (1) simplified patient-friendly summaries, (2) urgency level classifications, (3) multilingual translations, (4) audio narration via text-to-speech, and (5) contextual chatbot assistance for follow-up queries.

The platform is built on a FastAPI backend integrated with Google Gemini AI models and a React.js frontend with Tailwind CSS. A browser extension extending system reach to hospital portals via Chrome and Tampermonkey is also included.

The remainder of this paper is organised as follows. Section II presents the literature review. Section III formalises the problem statement. Section IV describes the proposed methodology. Section V details the implementation and experimental setup. Section VI discusses results. Section VII concludes the paper and Section VIII outlines future directions.

II. LITERATURE REVIEW

AI and NLP technologies have been widely explored for medi- cal document understanding and healthcare accessibility. This section reviews prior work spanning medical report summari- sation, urgency detection, conversational AI, and multilingual healthcare systems.

A. Medical Report Summarisation

Transformer-based models such as BERT [3] and GPT [4] have demonstrated strong performance in clinical text understanding and summarisation. These models extract salient findings from radiology and pathology reports and generate concise interpretations, achieving high accuracy in identifying abnormalities from clinical text.

B. Urgency and Disease Classification

Machine learning classifiers including Support Vector Ma- chines, Random Forest, and deep neural networks have been applied to medical triage and urgency classification [10]. These systems analyse symptoms, diagnostic observations, and clinical keywords to determine severity levels, enabling faster prioritisation of critical cases.

C. Radiology AI and Image Analysis

Deep learning models such as CNNs have been deployed for medical image classification and abnormality detection [1, 16]. However, most image-based systems focus on visual pattern recognition and do not address patient-facing explanation of textual radiology findings—a gap MedLens directly addresses.

D. Healthcare Chatbots

NLP-powered chatbot systems have been proposed for auto- mated healthcare assistance and symptom guidance [9]. While these systems improve basic accessibility, most offer gener- alised responses and lack the ability to ground answers in specific uploaded patient reports.

E. Summary of Gaps

Existing systems address individual components in isolation. Few integrate summarisation, urgency detection, chatbot inter- action, multilingual translation, and audio accessibility within a unified patient-centric platform. MedLens addresses this gap comprehensively.

III. PROBLEM STATEMENT

Medical reports produced from radiology imaging, pathology testing, and clinical examinations contain complex technical terminology difficult for patients and non-medical users to comprehend. Most patients depend entirely on healthcare professionals for interpretation, creating communication bot- tlenecks and delays.

Formally, given an uploaded medical PDF report R con- taining clinical findings in technical language, the objec- tive is to design a system M that: (a) extracts and pre- processes textual content T from R; (b) generates a sim- plified patient-friendly summary S; (c) classifies urgency $U \in \{\text{Low, Moderate, High, Critical}\}$; (d) translates S into target regional language L; (e) converts S to audio output A; and (f) enables contextual question answering $Q \to \hat{A}$ grounded in T , such that patient comprehension and health- care accessibility are maximised.

IV. METHODOLOGY / PROPOSED SYSTEM

A. System Overview

MedLens is designed as an end-to-end intelligent healthcare pipeline transforming complex medical reports into accessible patixient-friendly insights. The architecture integrates s mod- ules: Input Sources, Frontend Interface, Backend Processing, AI Features Engine, Browser Integration, and Output/Results Display.

B. System Workflow

Figure 1 illustrates the complete AI-driven processing workflow.

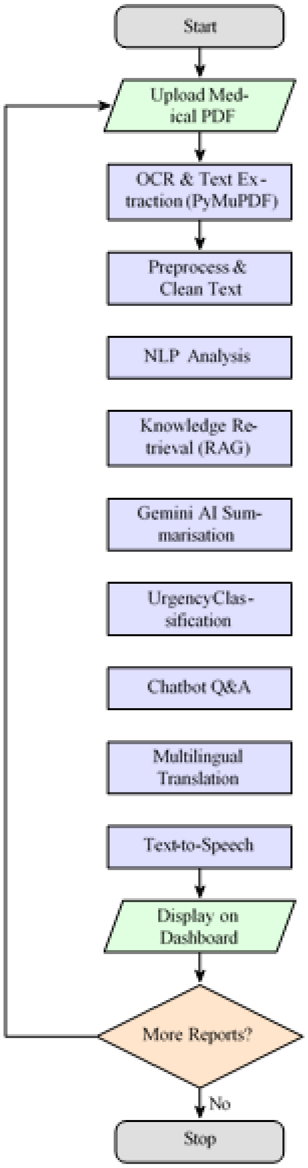


Figure 1: MedLens AI-Based Medical Report Analysis Workflow

C. Processing Stages

Stage 1 – OCR Extraction: PDF reports are processed using PyMuPDF to extract raw text from both digital and scanned documents. Key report sections (findings, impressions, observations) are identified and artefacts removed.

Stage 2 – NLP Analysis: Extracted text undergoes NLP- based processing to identify clinical entities, interpret medical terminology, and structure content for downstream AI modules.

Stage 3 – Knowledge Retrieval (RAG): Retrieval-Augmented Generation combines report-specific content with retrieved medical knowledge, improving accuracy and contextual reliability of generated outputs.

Stage 4 – AI Summarisation: Preprocessed clinical content is passed to Google Gemini AI, which generates simplified patient-friendly summaries with highlighted key findings and recommendations.

Stage 5 – Urgency Classification: The module scans for high-risk medical terms (e.g., tumor, malignant, bleeding, fracture) and assigns a severity level from {Low, Moderate, High, Critical} with a confidence score.

Stage 6 – Multilingual Translation: Summaries are translated into regional Indian languages (e.g., Kannada) using AI-based neural translation preserving medical context.

Stage 7 – Text-to-Speech: Simplified summaries are converted to natural speech, improving accessibility for visually impaired users, elderly patients, and low-literacy populations.

Stage 8 – Chatbot Assistance: A contextual chatbot grounded in extracted report content handles follow-up queries such as “Is this serious?” or “Which doctor should I consult?”, generating clinically grounded healthcare guidance.

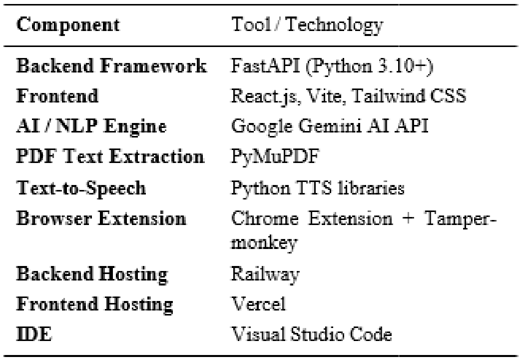
D. System Architecture

The architecture is structured across six layers: (1) Input Sources – PDF uploads from devices, browser tabs, or hospital portals; (2) Frontend Interface – React.js + Vite + Tailwind CSS dashboard; (3) Backend Processing – FastAPI + Python handling OCR, NLP, AI calls, and urgency detection; (4) AI Features Engine – Gemini AI for summarisation and chatbot, translation and TTS modules; (5) Browser Integration – Chrome Extension and Tampermonkey for direct hospital portal access; (6) Output/Results – structured summaries, urgency labels, translations, audio, and chatbot responses on the dashboard.

V. IMPLEMENTATION / EXPERIMENTAL SETUP

A. Software Stack

Table 1: Software Components of MedLens



Component	Tool / Technology
Backend Framework	FastAPI (Python 3.10+)
Frontend	React.js, Vite, Tailwind CSS
AI / NLP Engine	Google Gemini AI API
PDF Text Extraction	PyMuPDF
Text-to-Speech	Python TTS libraries
Browser Extension	Chrome Extension + Tampermonkey
Backend Hosting	Railway
Frontend Hosting	Vercel
IDE	Visual Studio Code

B. Hardware Requirements

The system runs on commodity hardware. A minimum of an Intel Core i5 processor, 8 GB RAM, and 256 GB storage is recommended. Cloud deployment via Railway and Vercel removes the need for specialised on-premise medical hardware.

C. Algorithm

1. Initialise system; accept user-uploaded PDF report R.
2. Extract textual content T from R using PyMuPDF.
3. Preprocess and clean T : remove artefacts, normalise whitespace.
4. Submit T to Gemini AI; generate patient-friendly summary S.
5. Scan T for high-risk keywords; assign urgency level U with confidence score c.
6. Accept chatbot query Q; generate report-grounded answer $\hat{A}$ from T .
7. Translate S into selected regional language L to produce SL.
8. Convert S to audio output A via text-to-speech synthesis.
9. Render S, U , c, SL, A, and $\hat{A}$ on the dashboard.
10. If additional reports remain, repeat from Step 1.

D. Testing

The system was tested on a diverse collection of medical PDF reports including MRI brain study reports, chest X-ray reports, pathology reports, and clinical diagnostic summaries. Evaluation was performed qualitatively by assessing: clarity of generated summaries, correctness of urgency classification, translation fidelity into Kannada, and chatbot response relevance to uploaded report content.

VI. RESULTS AND DISCUSSION

A. Text Extraction and Preprocessing

PyMuPDF-based text extraction performed accurately across both digital and scanned PDF reports. Extracted content was well-structured following preprocessing. Minor degradation was observed for very low-quality scanned documents with heavy noise or skewed text.

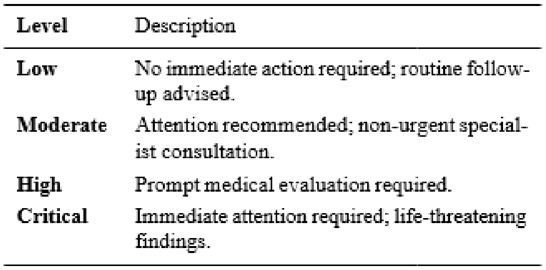
B. AI Summarisation

The Gemini AI model successfully converted complex medical language into concise patient-friendly summaries. Technical terms such as extra-axial mass, dural tail sign, and superior sagittal sinus occlusion were described in accessible language, enabling non-medical users to readily understand their findings.

C. Urgency Detection

The urgency classification module demonstrated accurate detection of critical medical conditions. In a representative brain imaging report test case, the system correctly classified findings as CRITICAL (confidence: 100%) and automatically flagged high-risk keywords including tumor, malignant, severe, and bleeding. Table 2 summarises the urgency classification categories.

Table 2: Urgency Classification Levels



Level	Description
Low	No immediate action required; routine follow-up advised.
Moderate	Attention recommended; non-urgent specialist consultation.
High	Prompt medical evaluation required.
Critical	Immediate attention required; life-threatening findings.

D. Multilingual Translation

Summaries were successfully translated into Kannada while preserving medical context. Users with limited English proficiency could read their report findings in their native language, validating the multilingual integration.

E. Text-to-Speech

The audio narration module produced clear, naturally spoken output. This feature proved particularly valuable for elderly users and visually impaired individuals.

F. Chatbot Assistance

The contextual chatbot accurately answered report-specific follow-up queries. When asked "Which doctor should I con- sult?" for a brain mass finding, the chatbot recommended consultation with a neurosurgeon or neurologist, grounding the response in the extracted report content. This approach significantly reduced generic or irrelevant outputs.

G. Browser Extension

The Tampermonkey extension integrated successfully with Chrome-based PDF viewers and hospital portals. Users could paste report text directly into the extension interface and re- ceive instant AI-generated summaries and urgency classifica- tions without requiring manual file downloads.

H. Deployment Performance

Cloud deployment via Railway (backend) and Vercel (fron- tend) provided stable real-time performance across concurrent report submissions, returning complete AI-generated insights within seconds per report. No specialised hardware was re- quired, confirming cost-effectiveness and scalability.

VII. CONCLUSION

This paper presented MedLens, an AI-powered radiology re- port simplification system designed to bridge the communica- tion gap between complex clinical documentation and patient comprehension. The system integrates PyMuPDF-based text extraction, Google Gemini AI summarisation, urgency classi- fication, multilingual translation, text-to-speech narration, and a contextual chatbot within a unified responsive web platform deployed on cloud infrastructure.

Experimental evaluation demonstrated that MedLens effec- tively simplifies radiology and clinical reports into patient- friendly language, accurately identifies critical medical con- ditions, supports regional language accessibility, and enables interactive conversational assistance. The system operates on standard hardware without specialised medical infrastruc- ture, making it practical and scalable for real-world healthcare deployment.

MedLens represents a meaningful step toward patient-centric digital healthcare, empowering individuals with clearer understanding of their medical conditions and facilitating more informed discussions with healthcare professionals.

VIII. FUTURE WORK

Several directions are identified for future enhancement of MedLens:

- Medical Image Analysis: Direct interpretation of X-ray, MRI, and CT scan images using computer vision and deep learning to complement textual report analysis.
- Mobile Application: Android and iOS applications for improved portability in remote and rural settings.
- EHR Integration: Seamless integration with Electronic Health Record systems for real-world clinical deployment.
- Extended Language Support: Additional Indian and inter- national regional languages to expand multilingual accessi- bility.
- Voice Interaction: Speech-based query input enabling hands-free access for elderly and physically impaired users.
- Advanced Conversational Memory: Persistent multi-turn AI with personalised health history context for improved chatbot assistance.
- Security and Compliance: Enhanced encryption and regu- latory compliance frameworks (e.g., HIPAA, DPDPA) for secure healthcare data handling.

REFERENCES

[1] A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter,
H. M. Blau, and S. Thrun, "Dermatologist-level classifi- cation of skin cancer with deep neural networks," Nature, vol. 542, no. 7639, pp. 115–118, 2017.
[2] E. J. Topol, "High-performance medicine: The con- vergence of human and artificial intelligence," Nature Medicine, vol. 25, no. 1, pp. 44–56, 2019.
[3] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for lan- guage understanding," in Proc.

NAACL-HLT, Minneapolis, MN, USA, 2019, pp. 4171–4186.

[4] T. Brown et al., "Language models are few-shot learners," in Proc. NeurIPS, 2020, pp. 1877–1901.

[5] Google DeepMind, "Gemini: A family of highly capable multimodal models," Google Research, 2024.

[6] A. Vaswani et al., "Attention is all you need," in Proc. NeurIPS, 2017, pp. 5998–6008.

[7] M. M. Rahman and Y. Wang, "Optimizing OCR accuracy for medical documents," International Journal of Computer Applications, vol. 177, no. 25, pp. 20–27, 2020.

[8] J. Cohen, "Artificial intelligence in radiology," Radiology, vol. 290, no. 2, pp. 318–321, 2019.

[9] D. Demner-Fushman, W. W. Chapman, and C. J. McDonald, "What can natural language processing do for clinical decision support?" Journal of Biomedical Informatics, vol. 42, no. 5, pp. 760–772, 2009.

[10] A. Holzinger, "Interactive machine learning for health informatics," Brain Informatics, vol. 3, no. 2, pp. 119–131, 2016.

[11] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.

[12] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in Proc. NeurIPS, 2013, pp. 3111–3119.

[13] OpenAI, "GPT-4 technical report," OpenAI Research, 2023.

[14] J. Hirschberg and C. D. Manning, "Advances in natural language processing," Science, vol. 349, no. 6245, pp. 261–266, 2015.

[15] R. Smith, "An overview of the Tesseract OCR engine," in Proc. ICDAR, 2007, pp. 629–633.

[16] P. Rajpurkar et al., "CheXNet: Radiologist-level pneumonia detection on chest X-rays with deep learning," arXiv preprint arXiv:1711.05225, 2017.

[17] D. Jurafsky and J. H. Martin, Speech and Language Processing, 3rd ed. Pearson Education, 2022.

[18] FastAPI, "FastAPI framework documentation," 2024. [Online]. Available: https://fastapi.tiangolo.com

[19] React Documentation Team, "React: A JavaScript library for building user interfaces," Meta Platforms Inc., 2024. [Online]. Available: https://react.dev

[20] PyMuPDF Documentation, "PyMuPDF: Python bindings for MuPDF," 2024. [Online]. Available: https://pymupdf.readthedocs.io